MULTIMEDIA UNIVERSITY

STUDENT ID NO

□□□□□□□□□□

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2018/2019

TPL 2141 – PROGRAMMING LANGUAGE CONCEPT

(All sections / Groups)

20 OCTOBER 2018
9.00 a.m – 11.00 a.m
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This Question paper consists of 5 pages (excluding cover page) with 5 Questions.
2. Attempt **ALL** questions. All questions carry equal marks and the distribution of the marks for each question is given.
3. Please write all your answers in the Answer Booklet provided

Question 1

(a) Given two sample codes of different programming languages.

```
<!DOCTYPE html>
<html>
<body>
<button type="button"
onclick="document.getElementById('demo').innerHTML = Date()">
Click me to display Date and Time.    </button>
<p id="demo"></p>
</body>
</html>
```

Language 1

```
(defun Hello (x)  (* x x))

(Hello 10)
(write-line "I am learning programming from the beginning")
```

Language 2

(i) In what way do Language 1 differ from Language 2? Discuss two (2) differences in terms of the language category and implementation method.

(ii) If you are requested to develop a program to solve a mathematical problem such as the enumeration of recursively defined factorial result, which of the above languages will you choose? Explain your decision based on the programming domains and writability criteria.

[2 + 2 = 4 marks]

(b) A Von Neumann architecture machine, designed by physicist and mathematician John Von Neumann (1903–1957) is a theoretical design for a stored program computer that serves as the basis for almost all modern computers.

(i) Describe the Von Neumann computer architecture with the aid of a diagram.

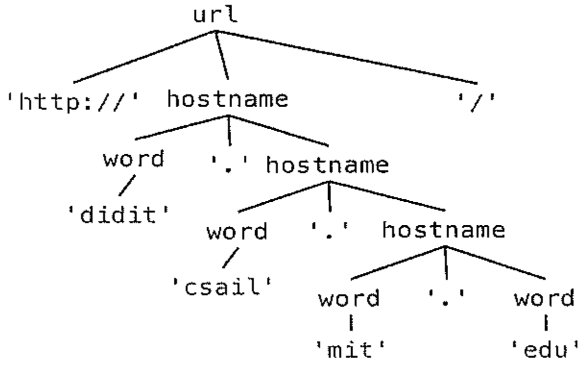
(ii) Explain why Von Neumann computer architecture has the strongest influences on the basic design of the imperative or procedural languages.

[4 + 2 = 6 marks]

Continued ....

Question 2

Consider the following parse tree:



(a) Provide the sentence that is derived from the above parse tree.

[2 marks]

(b) List out THREE (3) possible non-terminals from the above parse tree.

[1.5 marks]

(c) Construct the context-free grammar defined by the above parse tree.

[4 marks]

(d) Is the grammar ambiguous? Prove with this sentence: `http://www.mmu.edu.my/`

[2.5 marks]

Continued ....

Question 3

In **C++**, a programmer can declare several variables with the syntax: `int col, row;` This declaration can be described with the following grammar with start symbol 'decl' and terminals: `int, float, id.`

```
decl → type id_list ;
type → int | float
id_list → id_list , id | id
```

(a) Given the following attribute grammar for the above grammar.

Syntax Rules	**Semantic Rules**
decl → type id_list ;	id_list.type = type.value

id_list$_0$ → id_list$_1$, id	id_list$_1$.type = id_list$_0$.type
	id.type = id_list$_0$.type

id_list → id	id.type = id_list$_0$.type

(i) Discuss each of the semantic rules for the associated syntax rules.

(ii) Decide whether the attributes are synthesized or inherited. Explain.

[2 + 1 = 3 marks]

(b) Why is the above grammar only applicable for bottom-up parsing but not suitable for top-down parsing? Explain.

[2 marks]

(c) Modify the original grammar so that it can be utilized for top-down parsing. Show the steps.

[5 marks]

Continued ....

Question 4

(a) Given six symbols of the commonly-used relational operators:

>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
==	equal to
! =	not equal to

(i) How are these operators being used in a relational expression and what output is expected from a relational expression? Explain.

(ii) **JavaScript** and **PHP** have two additional relational operator, which are === and ! ==. Compare the behaviors of these additional operators with == and ! =.

(iii) Operator symbols may vary among different programming languages. Show TWO (2) different symbols but having the same behaviors.

[2 + 2 + 1 = 5 marks]

(b) The following pseudo-code demonstrates the effect of scoping on variable bindings:

```
a:integer
procedure first
      a:=1
procedure second
      a:integer
      first()
procedure main
      a:=2
      second()
      write_integer(a)
```

Discuss the process of finding the outputs based on the static and dynamic scoping during program execution.

[3 marks]

(c) The use of an operator for more than one purpose is referred as *operator overloading*, this incident is commonly applied in several languages. Show the example of operator overloading using the ampersand (&) as unary and binary operator in **C++** language.

[2 marks]

Continued ....

Question 5

(a) Given the following subprogram definition and call from **Ruby** language.

Subprogram definition:

```
def tester(p1,p2,p3,*p4)
      . . .
end
```

Subprogram call:

```
list = [2,4,6,8]
tester(‘first’,{mon=>72,tue=>55,wed=>70},*list)
```

(i) List out the formal and actual parameters.

(ii) After the binding of actual parameters to formal parameters, find out the parameters for `p1` and `p3`.

[2 + 2 = 4 marks]

(b) Analyse the following program from **C** language. After execution, this program generates the **same output** for the value of **x** according to different parameter-passing methods: passed-by-reference and passed-by-value-result. Explain the differences of the internal operations of each method based on the given program.

[4 marks]

```
void foo (int a, int b)
{
      a = 2 * b + 1;
      b = a - 1;
      a = 3 * a - b;
}
```

```
void main()
{
      int x = 5;
      foo (x,x);
      print (x);
}
```

(c) The commonly-used control structures in **R** language are `if, else, for, repeat, break, next, while` and `return`. Explain how did the control structures operating in the following program.

[2 marks]

```
for (i in 1:20) {
    if (i%%2 == 1) {
        next
    } else {
        print(i)
    }
}
```

End of Page